Doc Code: AP.PRE.REQ

PTO/SB/33 (07-05)
Approved for use through xx/xx/200x. OMB 0651-00xx
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

PRE-APPEAL BRIEF REQUEST FOR REVIEW

Docket Number (Optional)
POU920010121US1

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to "Mail Stop AF, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450" [37 CFR 1.8(a)]

on January 18, 2006

Signature Karen Taragowski

Typed or printed name Karen Taragowski

Application Number	Filed
09/944,721	8/31/01

First Named Inventor
Dennis A. QUAN, Jr.

Art Unit	Examiner
2179	Truc T. Chuong

Applicant requests review of the final rejection in the above-identified application. No amendments are being filed with this request.

This request is being filed with a notice of appeal.

The review is requested for the reason(s) stated on the attached sheet(s).
Note: No more than five (5) pages may be provided.

I am the

☐ applicant/inventor.

☐ assignee of record of the entire interest.
See 37 CFR 3.71. Statement under 37 CFR 3.73(b) is enclosed.
(Form PTO/SB/96)

☑ attorney or agent of record. Registration number 37,333

☐ attorney or agent acting under 37 CFR 1.34.
Registration number if acting under 37 CFR 1.34 _____

Signature

Jon A. Gibbons
Typed or printed name

(561) 989-9811
Telephone number

January 18, 2006
Date

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required.
Submit multiple forms if more than one signature is required, see below*.

☑ *Total of 1 forms are submitted.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appl. No.	:	09/944,721	Confirmation No. 1719
Applicant	:	Dennis A. Quan JR.	
Filed	:	08/31/2001	
TC/A.U.	:	2179	
Examiner	:	Truc T. CHUONG	
Docket No.	:	POU920010121US1	
Customer No.	:	23334	

PRE-APPEAL BRIEF REQUEST FOR REVIEW

The following remarks are submitted with the Applicant's notice of appeal. The references cited by the Examiner do not teach each and every element in the independent claims of the instant application, as required by 35 U.S.C. § 102.[1]

One embodiment of the present invention can be explained in terms of the word processing program such as Microsoft Word ("MS Word"). In MS Word, when a document is opened and a user types at least one character into the document, then the user selects the "File" menu list option at the top of the screen, a drop-down menu list will open that presents choices to the user such as "New," "Open," "Close," "Save," and "Print". See FIG. 3, elements 318-328 of the instant application. However, if a user has not yet typed any characters, the options "Save," and "Print" will not appear, because there is nothing to save or print. Further, if a user has opened MS Word, but there is no document open inside of MS Word, clicking on the "File" menu list at the top of the screen will not yield the "Close," "Save," or "Print" options, because there is nothing to close, save, or print.

This principle applies to other user-interface objects as well, such as tables within an MS Word document. If a table is inserted within in a document, clicking on the "Table"

[1] See MPEP §2131 (Emphasis Added) "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." Verdegaal Bros. v. Union Oil Co. of California, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). "The identical invention must be shown in as complete detail as is contained in the ... claim."

menu list will yield options such as "Insert Rows" or "Delete Rows." If no table is in the document, those options would not be applicable and should not be displayed.

In the prior art, programmers had to assign all of these attributes to each state of a program (i.e., document open, document open with text, etc.) during programming. The present invention allows the state of the program to dynamically determine the available options, thereby avoiding the tedious and time consuming task of programming each state.

In the present invention, a user interface (UI) object, such as a document, has a menu list associated with it. The menu list has choices, such as "File", "New", "Open", "Close", etc. See FIG. 3. The novelty resides in the fact that each choice is associated with an "associative array" (see items 306-310 in FIG. 3). The term "associative array" is a well-known term of art in the computer software field and is defined on page 11 of the specification of the present application as "*a set of items, which are randomly accessible by a key, often a string.*"

The particular choices available to a user within each menu vary dependent upon a Boolean matching of the associative array entries to global context flags assigned to each document or object within the document, such as a table. These global context flags 304 are nested in structure and usage as shown in FIG. 3 and described on page 11, lines 21-22. Furthermore, as is shown in Table 3 and described in the specification, the global context flags are also associative arrays and the present invention is logically comparing two associative arrays using a Boolean AND operator. Page 11, line 18 through page 12, line 12.

Returning to the MS Word example, when the application is running, and no document is open, (See FIG. 3) the global context flags are set to document=closed, document type=none, and document object=none. These are presented to three possible menu lists, in this example, file 318, table 330, and graphic 338. Each list has an associative array attached, which defines when the object should be shown. By logically comparing

the associative array forming the global context flags with the corresponding menu items in an associative array using a Boolean AND function, the menu list of items that should be shown to a user are computed. For instance, if a document isn't opened, it wouldn't make sense to present a user with an option to "Save."

Hobbs Does Not Teach Boolean Comparison of Two Associate Arrays

Independent claims 1 and 11 each recite performing a Boolean comparison of two associative arrays, i.e., global context flags (as defined in the spec as an associative array) and the associative array of UI objects. Specifically, claims 1 and 11 recite "performing a Boolean comparison between the global context flag and one or more of the entries in the associative array for each of the UI objects". Clearly this is not taught by Hobbs. Accordingly, independent claims 1 and 11 distinguish over Hobbs for at least this reason and the Examiner's rejection should be withdrawn.

Even if Applicants don't rely on the definition of global context flags as recited in the specification, claim 18 positively recites the Boolean comparison of two associative arrays. Specifically, claim 18 recites "performing a Boolean comparison between: at least one global context flag in the first associative array; and at least one entry in a second associative array". The global context flags are defined by the state of the UI object to which they pertain. Refer to the present invention at page 11, line 18 through page 12, line 3. Accordingly, independent claim 18, distinguishes over Hobbs for at least this reason and the Examiner's rejection should be withdrawn.

The Hobbs reference cited by the Examiner discloses a browser that passes a request, including an argument symbol, to an Application Server. Hobbs, col. 27, lines 59-61. On pages 2 and 3 of the June 24, 2005 Office Action the Examiner compares the menu of choices in the window display of Hobbs to the UI objects of the present invention and the window in Hobbs to the UI of the present invention. The Examiner also compares the "argument that acts as a key" in Hobbs to the global context flags of the present invention.

The argument symbol of Hobbs is used to find an address to a database or internet site where the requested information resides. The disclosed methods of locating the address include Boolean searching for the argument in a look-up table, a hash table, an associative array, or a linked list. Hobbs, col. 11, lines 56-58. In the case of the associative array, the argument is used as a key. Hobbs, col. 18, lines 4-6. Importantly, Hobbs does not disclose, teach or even suggest Boolean matching two associative arrays, but is instead Boolean searching within a single array for a key word that is an address in a database or to an internet site. Accordingly, independent claims 1, 11, and 18, distinguish over Hobbs for at least this reason and the Examiner's rejection should be withdrawn.

Hobbs Does Not Teach Altering The State of the Argument or Key

The text in Hobbs cited by the Examiner states that "the key can be used to create a window display for viewing by a user, the window presenting the user with a menu of choices for further areas of research pertaining to the key. The user, by selecting one of the menu choices, causes the application...to match a key..." Hobbs also states that, "each choice corresponds to an Argument Symbol." Hobbs, col. 18, lines 25-27. In other words, each choice is a new argument and thus, a new key. However, the "argument" in Hobbs is simply a static link to an HTTP network address. The argument never changes. Nowhere does Hobbs mention a state of the key, as do claims 1 and 11. The key in Hobbs never changes from being a link to a specific address. Therefore, the Examiner has not shown and is unable to show that Hobbs "present[s] at least one UI object based on a state of at least one global context flag for the UI," as recited in independent claims 1 and 11 of the instant application. Accordingly, independent claims 1 and 11 distinguish over Hobbs for at least this reason and the Examiner's rejection should be withdrawn.

In addition, the Examiner has not shown that Hobbs discloses "altering the state of the global context flag based on the response from the end-user," as recited in independent claims 1 and 11 of the instant application. Again, Hobbs is completely silent on altering the state of anything. In Hobbs, a user selects a hyperlink and, in one embodiment,

makes a second choice of further areas of research pertaining to the hyperlink chosen. Hobbs, col. 18, lines 22-30. No states are changed or altered. The further choices are new arguments and, thus, new keys to search with. Therefore, even if, arguendo, the key in Hobbs is similar to the global context flag of the present invention, the Examiner has not shown where the state of the key has been altered. Accordingly, independent claims 1 and 11 distinguish over Hobbs for at least this reason and the Examiner's rejection should be withdrawn.

Continuing further, when the linked term in Hobbs is selected by a user, such as linked term 303 in FIG. 6 of Hobbs, the browser sends a request to the Application Server. "The Application Server responds by outputting HTML...elements...that are described in the CGI application on the Application Server" Hobbs, col. 19, lines 24-49. In other words, once the hyperlink is clicked on, other window, links, text, etc. are placed on the screen. There is no longer any motivation to show the same hypertext to a user. Therefore, even if, arguendo, the hyperlink in Hobbs is a UI object, the UI object is not updated but is instead replaced with new links or web pages. Accordingly, the Examiner has not shown that Hobbs "updates" the UI objects, as recited in independent claim 18 of the instant application. Accordingly, independent claim 18 distinguishes over Hobbs for at least this reason and the Examiner's rejection should be withdrawn.

In view of the foregoing, independent claims 1, 11, and 18 distinguish over Hobbs because one or more elements are not present in Hobbs. All the remaining claims i.e. 2-10, 12-17, and 19-23 depend respectively from independent claims 1, 11, and 18. Accordingly, the claims 1-23 of the present invention distinguish over Hobbs for the reasons shown above. The Applicant respectfully requests that the claims 1-23 of the present invention be allowed or in the alternative reopen prosecution on the merits citing art teaching every element recited in the claims.

Respectfully submitted,

Date: January 18, 2006

By: _____
Jon Gibbons, Reg. No. 37,333
Attorney for Applicants